Image-Based Window Detection – An Overview

Neuhausen M.*, Koch C.**, König M.*
*Ruhr-University Bochum, **The University of Nottingham
marcel.neuhausen@ruhr-uni-bochum.de

Abstract. The automated recognition and segmentation of building façades and the detection of their elements are of high relevance in various fields of research as they, e.g., reduce the effort of 3D reconstruction of existing buildings and even entire cities or may assist in navigation and localization tasks. In recent years, several approaches were made concerning these issues. They can be mainly classified by their type of input data which are either individual images or 3D point clouds. This paper provides a survey of image-based approaches. Particularly, this paper focuses on window detection and, therefore, groups related papers into three major detection strategies. We juxtapose grammar-based methods, pattern recognition algorithms and machine learning techniques and contrast them with regard to their generality of application. As we found out, machine learning approaches seem most promising for window detection on generic façades. We will pursue these ideas in future work.

1. Introduction

In recent years, the Building Information Modelling concept of using 3D building models for construction and maintenance purposes has increasingly developed towards creating models of entire cities as civil engineering goes beyond the scope of single buildings. Moreover, models of existing buildings are required in other application areas such as virtual and augmented reality, navigation and localization, or simulation.

Today, 3D city models often are publicly available from services like Google Earth or can be requested from land-registries. However, since such models are commonly released as coarse block models, they often lack details, such as the shape of roofs, façades, or façade elements. For most applications, these models are insufficient since façade elements constitute the major part of a building. The richness of detail strongly matters in virtual reality applications to guarantee the effect of emersion. Furthermore, façade elements and windows in particular play a major role for recognizing buildings in terms of navigation and localization, e.g., tourist guidance (Ali et al., 2007), as they essentially contribute to a distinctive buildings’ design. Similar aspects apply to other, civil engineering related, tasks like simulating the stability of a building. Especially mechanized tunneling projects may benefit from more detailed models as they assist risk assessment of settlement-induced damages inflicted to urban structures (Obel et al. 2016). In particular, the so-called wall-openings coefficient, meaning the ratio of opening sizes to total wall area, is a crucial factor in risk assessment. Openings in a solid wall decrease its stiffness and, thus, increase the vulnerability to settlements. Windows, in this regard, account for the major part of wall-openings.

Façade reconstruction including window detection is, hence, a crucial issue which has not yet been solved sufficiently (Musialski et al., 2013). In this paper, we present an overview of the major strategies for automated window detection on façades and discuss their advantages and limitations regarding tunneling projects.
2. Related Work

In the past decades an extensive body of literature arose concerning building reconstruction and particularly façade reconstruction. The approaches made in this field of research are highly diverse and manifold. However, some effort to categorize existing methods has already been made. Usually approaches are differentiated by the type of input data. Baltavias (2004), Brenner (2005), and Haala and Kada (2010) provide surveys focusing on airborne imagery and laser scans. Although rudimentary building reconstruction is possible, this data source is not suitable for façade elements recognition. Baltavias (2004) and Brenner (2005), hence, address the detection of large structures like buildings and streets and the reconstruction of simple building shapes. On the contrary, Haala and Kada (2010) additionally include approaches using terrestrial laser scans to gain more detailed geometrical information. They also present approaches developing façade texture mapping to reconstructed 3D models of buildings. In the comprehensive survey of approaches on building and façade reconstruction of Musialski et al. (2013), the authors cover both airborne and terrestrial imagery and laser scans. Additionally, they differentiate between automatic and semi-automatic approaches. Although they discuss approaches of pattern detection, matching as well as façade parsing, the paper focuses on reconstruction, while façade element detection is only mentioned briefly.

3. Approaches

In the following, we concentrate on approaches based on terrestrial imagery, as we focus on window detection in this paper. Indeed, laser scan generated point clouds simplify the detection of façade elements and especially windows (van Gool et al., 2007). However, the acquisition is very time-consuming and expensive as risk assessment in tunneling projects requires models of entire cities or at least models of areas along the tunneling alignment. Façade images, contrariwise, are mostly publicly available from Google Street View and other services or may be easily gathered manually or even semi-automatically by drone flights as Freimuth and König (2015) have shown.

Most approaches using ground view imagery can be roughly categorized into three main strategies, although some approaches combine multiple strategies, which interdicts a clear disjunction in general. However, for comparison reasons the following partition by strategies facilitates a contrasting juxtaposition. We classify multiple strategy approaches by their prominent method.

The remainder of this paper is structured as follows: section 3.1 gives an overview of approaches applying grammars to façade images. Section 3.2 summarizes pattern recognition approaches and section 3.3 relates to window detection using machine learning techniques. In section 4 we discuss the limitations of these strategies with respect to their generality of application. Finally, section 5 concludes our findings and makes proposals for further research.

3.1 Grammar-Based Approaches

The most prominent methods in grammar-based façade reconstruction are shape and splitting grammars. These make use of assumptions regarding symmetry, such as the grid-like arrangement of façade elements or their usually rectangular shapes. In the context of several comprehensive publications Ripperda and Brenner (2009) proposed an approach applying a formal grammar to façades for automated reconstruction. Exploiting the assumptions mentioned above, Ripperda (2008) designed a context-free grammar splitting a façade into its symmetric and repeating areas. To optimize the grammar’s segmentation of a façade, the
authors proposed a reversible-jump Markov Chain Monte Carlo (rjMCMC) method sampling over numerous runs (Ripperda and Brenner, 2006). Thus, façade elements are automatically derived. To improve the sampling, Ripperda and Brenner (2007) use measurements of their data to control the sampling process which leads to a reduction of false hypotheses. Müller et al. (2007) proposed an approach which splits the façade into floors and tiles by means of symmetry information. Ensuing, it subdivides the tiles into smaller rectangular regions and organizes them in clusters. The extracted architectural elements can then be matched with 3D objects in a database. For façade reconstruction, the windows’ depth information has to be explicitly specified via interaction with users. Compared to this, Teboul et al. (2010) defined a more detailed shape grammar that does not only split a façade into rectangles but models semantic relationships of façade parts and their elements, e.g., the relationship of an attic to a roof or a window. The authors understand façade segmentation by use of a grammar as finding a specific sequence of rules. Thus, they use an energy minimization scheme determining the best fitting manifestation on particular images. In a last step, they classify each segmented part of the image by a randomized decision forest which labels each pixel according to its probability of class affiliation. In a subsequent approach, Teboul et al. (2013) are content with a simple shape grammar that only allows iteratively splitting rectangles in smaller ones in order to focus on improving the parsing of façades. The parsing is then done by a reinforcement learning algorithm. Like the approach of Ripperda and Brenner (2006), many other approaches draw on sampling algorithms like MCMC for the optimization of parse trees. Riemenschneider et al. (2012) pointed out that sampling such a large solution space holding a complex structure is not a sufficient method. For this reason, they propose the Cocke-Younger-Kasami algorithm (CYK) for efficiently parsing context-free grammars. In their approach the authors define a grammar allowing for irregular lattices splitting the façade into rectangular regions of different sizes. Similar to Teboul et al. (2010), they assign the most probable label of class affiliation to each pixel using a merit function that describes the likelihood of each pixel belonging to a given class.

To stochastically sample parse trees or to find the most probable parse tree of an ambiguous grammar for a given input, probabilistic grammars are the method of choice. In contrast to deterministic grammars, these extend the former by stochastic components. Each production rule in a grammar’s set of rules additionally possesses a likelihood of appearance that, as a consequence, induces a probability distribution on the generated output. Assuming façades to be structured highly regularly and symmetrically as can be found at large office buildings, Alegre and Dellaert (2004) developed a stochastic context-free grammar which contains only horizontal and vertical splits of the façade. This is used to derive a Bayesian stochastic model which depicts a hierarchical partitioning of disjoint blocks of façade images. For approximating the posteriors of partitions, the authors apply MCMC sampling. Tyleček and Šára (2012) proposed a novel approach by relinquishing to specify a grammar with an explicit set of production rules. Instead they develop a stochastic model in a tree-like structure acting as a grammar. For optimizing the grammar’s manifestation, they use a general rjMCMC framework.

3.2 Pattern Recognition

As windows are mostly assumed to be rectangles arranged in a grid-like manner parallel to the façade’s contour, some approaches detect windows as patterns of horizontal and vertical edges in façade images. Lee and Nevatia (2004) proposed an approach to window detection based on rectified façade images, which was later resumed by Meixner et al. (2011) for interpreting façade elements. In the image, they project horizontal and vertical edges and superimpose resulting histograms with each other. Window candidates are assumed to be
located at the emerging peaks’ locations. Their position is finally verified based on the previously extracted edges. Since this method only detects windows of rectangular shape, Lee and Nevatia (2004) extended their approach to arch windows by allowing the detected window’s top edge to become curved and again verifying it with the image’s edges. To apply window detection to more irregular façades, Recky and Leberl (2010) picked up Lee and Nevatia’s (2004) concept. They deduce a subdivision of the façade into levels from a vertical edge projection of the façade image and subsequently apply a horizontal edge projection separately on each level to further divide the levels into blocks. By applying a $k$-means clustering by color to these blocks, Recky and Leberl (2010) identify blocks which belong either to the façade or to a window. Final window detections are obtained by joining adjacent window blocks. Korah and Rasmussen (2008) discussed a method that focuses more on the grid-like alignment of windows on a façade. Therefore, they hypothesize rectangles from all pairs of parallel edges in the image resulting in a large quantity of hypotheses. As windows are supposed to be aligned in a grid, the best possible lattice of a subset of the extracted rectangles is constructed by means of a MCMC optimization procedure.

Instead of explicitly assuming the shape of windows to be rectangular, the *implicit shape models* (ISM) introduced by Leibe et al. (2004) combine the recognition and segmentation of objects in one process. These models consist of a codebook prototypically defining local features and a probability distribution determining their spatial positions. Reznik and Mayer (2008) applied ISM for window detection. Interest points in combination with their geometric arrangement around the window’s center constitute the ISM which is then learned from a manually labeled training set. Reznik and Mayer (2008) generate window hypotheses by cross correlating image patches around interest points with the training data. Retrieved good hypotheses are used in a self-diagnosis manner to validate weak hypotheses. For refinement of window alignments, model selection is done based on the assumption that windows are arranged in rows and columns.

Typically, façade elements occur recurrently on façades. Thus, the element alignments on façades can be understood as repetitive patterns. Van Gool et al. (2007) proposed an approach to window detection as a subtask of façade reconstruction that relies on repetitions on the façade. They tackle the problem by a bifid strategy approach depending on the quality of the input images. As for their method of fully automatic reconstruction, sufficient perspective effects in the images are required, otherwise they switch to the approach of Müller et al. (2007) (see section 3.1). If sufficient perspective is available, interest points occurring repeatedly over the façade will be computed and grouped to infer the camera’s calibration. Window detection is then done by energy minimization using graph-cut optimization. In a concluding refinement step van Gool et al. (2007) use shape priors for vertical and horizontal alignment.

### 3.3 Supervised Machine Learning

In supervised machine learning in the context of computer vision, object recognition is achieved by training a classifier using image features of a labeled data set. By training, classifiers learn a function to separate object classes. In this field, *support vector machines* (SVM) build a kind of algorithms able to deal with high dimensional inputs are widely deemed to yield good classification results (Yang et al., 2012). Inspired by Lee and Nevatia (2004) and Recky and Leberl (2010), Haugeard et al. (2009) utilize the fact that windows are mostly aligned to floors. They divide façades into floors by thresholding vertical edge histograms. On each floor they project horizontal edges to a histogram to infer window candidates. Edges of these candidates are extracted from the image and represented in a graph. Windows are then classified by a SVM using a kernel operating with inexact graph
matching. As a result, windows emerge as a sub-graph of the graph of all contours extracted from the façade image (Musialski et al., 2013).

Bag-of-visual-words (BoW) approaches offer another way to handle multiple different image features at once by integrating them into sparse vectors of feature occurrences. These can then be fed into a classifier. Csurka and Perronnin (2008) described a method making use of BoW for a pixel-wise semantic segmentation of façade images into façade elements. The approach relies on the detection of image patches out of which local image features are extracted and combined to a BoW representation. Classified by sparse logistic regression, the results are assigned back to the level of pixels. Fröhlich et al. (2010) improved this method. Instead of extracting image patches they compute local color features obtained by dense sampling. Classification is then done by feeding the gained feature vectors into a randomized decision forest.

The previously described classification approaches rely on high dimensional input vectors. Feature responses have to be explicitly arranged in vector shape. Boosting algorithms contrariwise allow for deriving a strong classifier directly from image features. Originally designed for face detection, Viola and Jones (2001) illustrated a boosted cascade of simple features as a method for object detection which uses image features as classifiers. Features are treated as weak learners and are fused by boosting in a way that they form a strong classifier. Ali et al. (2007) applied the approach of Viola and Jones (2001) to detect windows on façades. Divers thresholded Haar-like features (Oren et al., 1997) on different positions within the image act as a pool of weak learners out of which an adapted AdaBoost algorithm (Freund and Schapire, 1997) selects a subset and arranges them in cascading stages to form a strong classifier. For detection a search window is slid over the image and resulting patches are presented to the classifier. To enable detection of different façade elements Drauschke and Förstner (2008) described an approach similar to the one of Ali et al. (2007) which also builds a strong classifier by AdaBoost. Though, the novel idea is providing different kinds of image features at once to the AdaBoost algorithm. Since the detection and classification of façade elements using supervised learning algorithms are highly dependent on the features used, Yang et al. (2012) conducted an empirical study on feature evaluation in this context. For region-wise labeling the façade’s elements they use randomized decision forests based on different features as classifiers. Martinović et al. (2012) presented a similar approach like Yang et al. (2012) but in order to substantiate classification results, they augment the decision tree’s output with further processing results. In their three-layered approach they make use of an oversegmentation of a building’s façade image. In a first step, a recursive neural network merges oversegmented regions into objects. These hypotheses are then combined with the classification results of a decision tree of integral channel features (see Dollar et al., 2009) learned by a discrete AdaBoost. Finally, the authors refine detections by exploiting weak architectural principles, e.g., vertical and horizontal alignment or co-occurrence of objects.

It may be necessary to detect more than one single object class. In terms of façade element recognition this could mean distinguishing between windows, doors, and maybe even more categories. Fidler et al. (2006) established an approach to multiple class categorization based on hierarchical models reducing the computational complexity which such a task usually implicates. Mačák and Drbohlav (2011) adapted the generic approach given by Fidler et al. (2006) for the use of window detection. They applied a difference of Gaussians to extract edge features forming the lowermost layer of a hierarchical window model. From there on, following layers are learned such that the respectively superior layer is a composition of instances in local neighborhoods of the underlying layer. The resulting model is then used to detect windows in façade images.
4. Discussion

To apply window detection approaches to buildings of different architectural style or even to cities in various countries, the methods should generalize well as façade and window appearances can vary both in geographical region and in year of construction. In the following, we discuss the assumptions and limitations of the methods presented above.

**Grammar-Based Approaches.** For all grammar-based approaches an optimization of the parse tree is inevitable for each presented façade. This is a time-consuming procedure if applied to a high quantity of façade images, as is the case in our research. Optimization may either be done by sampling over all possible trees or by relying on special parsing algorithms like CYK. Especially sampling methods limit the field of application as they are, on the one hand, only applicable to grammars consisting of small rule sets and, on the other hand, can potentially get stuck in local optima (Riemenschneider et al., 2012). This mostly leads to simple shape grammars consisting of few rules which only split façade images into rectangles as windows and other façade elements are often assumed to be of rectangular shape. On the contrary, defining more complex grammars may be highly dependent on expert knowledge of building construction. This seems advantageous as there are explicit universal constructions rules. The aesthetic sensibility, however, differs in country which complicates deducing general rules. Moreover, the more a grammar’s set of rules is refined the stronger it gets biased to specific façade types. Thus, it is no longer generally applicable as architectural styles vary not only among countries but also among geographical regions and even the year of buildings’ construction (Haala and Kada, 2010). However, in case of simple shape and splitting grammars as well as in case of complex grammars, these approaches only achieve a rough segmentation of façade images. This has to be processed in further steps to both gain proper object contours and determine the object classes using either image processing or machine learning techniques. Since a pre-processing of the images is often required to obtain image features like symmetry or edge information before the grammar and the sampling algorithm can be applied, it is questionable if a detour over grammars yields a promising solution.

**Pattern recognition.** Approaches focusing on pattern recognition usually rely on a couple of assumptions. Assumptions on the distribution of pixel intensities, meaning that windows are darker than the surrounding façade, often lead to erroneous results (Musialski et al., 2013). Other approaches make assumptions referring to both the window shapes and their alignment on the façade. More precisely, like in the approach of van Gool et al. (2007), it is often anticipated that windows occur on façades as multiple elements of the same type, i.e. they have same size and shape, and repeat in fixed displacements in both horizontal and vertical direction. This practice yields remarkably good results as long as façades are highly regular structured as well as broad enough so that façade elements form the presumed patterns. This may be valid for structures like large office buildings but it does not hold for the majority of structures in typical European inner city areas and adjacent regions. In apartment buildings the windows of staircases often deviate in size from residual windows and also strongly break the pattern spanned by the latter. Detection of such asymmetric patterns cannot be well handled as van Gool et al. (2007) and Meixner et al. (2011) pointed out. Furthermore, most buildings are rather small resulting in less repetitions which also complicates detection. Another drawback Reznik and Mayer (2008) highlighted is that the images’ resolution and partial occlusions of windows by trees, traffic signs and alike highly affect the detection’s quality. This comes into account if publicly available images are used, such as from Google Street View, as these images are, firstly, usually taken from the street sight such that occlusions occur unavoidably and, secondly, are rectified whereby block artefacts may arise and lower the resolution.
Supervised Machine Learning. In contrast to the aforementioned strategies, supervised machine learning approaches rely neither on assumptions about the window alignments on the façade as they detect an object by its inherent characteristics instead of the interplay with other objects, nor do they need explicit prior knowledge about construction details. This is advantageous as both restrictions result in a loss of universal applicability. Nevertheless, the detection rate of machine learning algorithms is, amongst others, constrained by the type of image features and corresponding responses. Features extracted from the input images, thus, should be significant and relevant. This constitutes a challenge in itself, as windows usually expose rather few outstanding features, apart from their frames. However, there are already approaches like Yang et al. (2012) to cope with this issue. Certainly, for training a supervised machine learning algorithm a labeled set of training images is required which is time-consuming in gathering and setting up correctly. Since these algorithms, however, generalize well, this procedure has to be done only once to detect windows in façades of different construction years or architectural styles.

5. Conclusion

We highlighted the meaning of window detection to improve existing 3D city models for the use in civil engineering tasks such as tunneling. By comparing several approaches of different strategies, we discussed application limitations and concluded that there is, by now, no sufficient approach for universal use. Yet we conclude that machine learning techniques may yield promising results in future since they avoid restricting assumptions and offer an excellent ability of generalization.

We, thus, will focus on machine learning techniques to detect windows in our future work. Particularly, we believe that the approach of Ali et al. (2007) may be developable. An extension of this approach by a pre-processing of the image easing the feature extraction and unifying the windows’ characteristics as well as a post-processing to reject misclassification may increase the detection rate. Moreover, we will adapt the approach by substituting Viola and Jones’ (2001) detector by a Soft Cascade detector (Bourdev and Brandt, 2005) as this may also increase the detection rate. For training and testing issues we recommend the eTRIMS data set (Korč and Förstner, 2009) as it provides extensive samples and has already been used in several approaches which supports the comparability.

Currently, mostly edges features are used in window detection, in other respects, windows seem to be poor of features. In addition to Yang et al.’s (2012) work, we consider a further evaluation of image features to be desirable since in most façades there are other objects that provide even stronger responses to edge features (Recky and Leberl, 2010). Thus, finding sets of features which significantly describe the window characteristics is a major task. Features found in this context can also improve the results of the above mentioned approach.

In recent years, deep neural networks (DNN) have attracted more and more attention in computer vision for object classification tasks. Szegedy et al. (2013) proposed a method to not only classify objects but also locate them within an image. As such approaches often outperform classic computer vision algorithms, it would be desirable to further pursue these concepts and apply them to window detection.
Acknowledgement

Financial support was provided by the German Research Foundation (DFG) in the framework of the Collaborative Research Center SFB 837 “Interaction Modelling in Mechanized Tunneling”.

References


